Real-time Implementation and Evaluation of an Adaptive Energy-aware Data Compression for Wireless EEG Monitoring Systems

Alaa Awad[1], Medhat Hamdy[1], Amr Mohamed[1] and Hussein Alnuweiri[2]
[1]Department of Computer Science, Faculty of Engineering, Qatar University, Doha, Qatar
[2]Department of Electrical and Computer Engineering, Texas A&M University, Doha, Qatar
E-mail: {aawad, mhamdy, amrm}@qu.edu.qa and alnuweiri@gmail.com

***Abstract*—Wireless sensor technologies can provide the leverage needed to enhance patient-caregivers collaboration through ubiquitous access and direct communication, which promotes smart and scalable vital sign monitoring of the chronically ill and elderly people live an independent life. However, the design and operation of BASNs are challenging, because of the limited power and small form factor of biomedical sensors. In this paper, an adaptive compression technique that aims at achieving low-complexity energy-efficient compression subject to time delay and distortion constraints is proposed. In particular, we analyze the processing energy consumption, then an energy consumption optimization model with constraints of distortion and time delay is proposed. Using this model, the Personal Data Aggregator (PDA) dynamically chooses the optimal compression parameters according to real-time measurements of the packet delivery ratio (PDR) or individual users. To evaluate and verify our optimization model, we develop an experimental testbed, where the EEG data is sent to the PDA that compresses the gathered data and forwards it to the server which decompresses and reconstructs the original signal. Experimental testbed and simulation results show that our adaptive compression technique can offer significant savings in the delivery time with low complexity and without affecting application accuracies.**

***Index Terms*—Wireless healthcare applications, EEG signals, Cross-layer design, Convex optimization.**

I. Introduction

Recent advances in signal processing and low-power wireless communications have stimulated great interest in the development of wireless technology in biomedical applications, including wireless body area sensor networks (BASNs). BASNs boost the opportunity for remote monitoring of personal health. Remote diagnosis provides healthcare services for people residing in remote areas and people with chronic diseases, hence it can increase early detection of emergency conditions [1]. However, BASN has its peculiar design and operational challenges, particularly focusing on energy and delay optimization. Delay constraint and energy consumption of sensor nodes are fundamental design parameter for BASN. In this paper, we consider a wireless BASN consists of small and low-power sensors (e.g. electroencephalogram (EEG) sensors), and a resource-rich Personal Data Aggregation (PDA) device (such as a smartphone). These sensor nodes periodically send sensed information to a healthcare provider through the PDA.

Many approaches have been devised to address delay and energy minimization problems in BASN. However, most of these works have focused on energy efficient MAC protocols, by avoiding idle listening and collision [2], or by presenting latency-energy optimization [3][4]. For example, the authors in [3] presented a traffic-aware dynamic MAC (TAD-MAC) protocol. This protocol optimizes the energy consumption by reducing idle listening and unnecessary wake-up beacon transmission. In this context, every node adapts its wake-up interval dynamically according to the amount of traffic. The authors in [5] discussed the problem of minimizing time average energy consumption using transmission scheduling with a worst-case delay requirement. In this approach, the authors utilized both sleeping mode and opportunistic transmission for increasing energy efficiency. Furthermore, enhancing access protocol to minimize message delivering latency has been addressed in [6], and duty cycling/sleep scheduling schemes have been studied in [7]. To the best of our knowledge, most of the presented work that study the BASNs do not take both the encoding and transmission energy consumption into consideration [8]. Furthermore, the cross-layer design for energy minimization that addresses the time-frequency allocation under delay and distortion constraints, has not been studied before.

In this paper, we propose an adaptive compression framework that dynamically optimizes and adapts the encoder's parameters, according to real-time packet delivery ratios (PDRs), to achieve low-complexity energy-efficient compression, with constraints on end-to-end delay, Bit error rate (BER) and source coding distortion. First, we analysis encoder's parameters and processing energy consumption, then an optimization model that aims at minimizing the processing energy consumption, under an end-to-end delay deadline constraint, and a maximum allowable signal distortion constraint, is proposed. Then, we mathematically convert this model into a problem of geometric programming (GP) [9] and then solve it by cvx [10] to obtain the optimal solution. In addition to that, we

This work was made possible by NPRP grant # 09-310-1-058 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

QSHINE 2014, August 18-20, RHODES, Greece

DOI 10.4108/icst.qshine.2014.256334

developed a testbed to realize the proposed framework. This real-time implementation illustrates that the adaptive compression at the network level offers significant savings in the delivery time without affecting the application's classification accuracies and with minimum complexity.

The rest of the paper is organized as follows. Section II introduces the network model and the analyze of energy consumption and delay time calculations. Section III presents the optimization problem formulation. Section IV introduces the implementation of the proposed real-time system. Section V presents the simulation environment and results. Finally, Section VI concludes the paper.

II. Network Model and Problem Formulation

A. Network Model

In this paper, a wireless BASN, as shown in Figure 1 is considered. In this model, the Personal/Patient Data Aggregator (PDA) gathers the data from N sensor nodes pre-attached to it, and forwards the aggregate traffic to a central server. There is no possibility of collision within the network, as each sensor node has its own time and frequency slots.

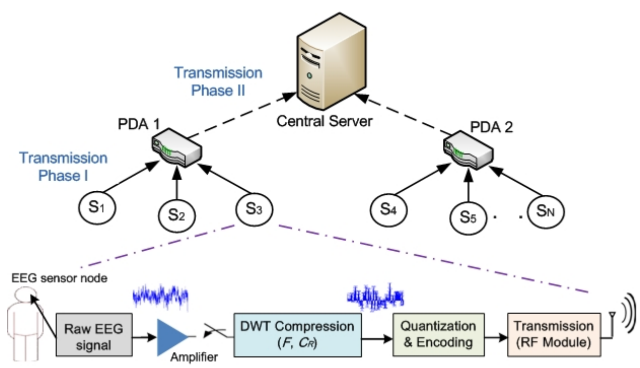

Fig. 1. System Model.

The proposed framework utilizes the encoding model of EEG signals. However, without loss of generality, the proposed model can be extended to a range of vital signs which are typically at a low data rate e.g., temperature, pressure or heart-rate reading, or at higher data rate such as streaming of ECG signals. The EEG signal is considered as the main source of information to study human brain, which plays an important role in diagnosis of brain disorders [11]. The importance of EEG signal also appears in Brain Computer Interface (BCI) applications [12]. The general structure of the typical-used EEG encoder is illustrated in Figure 1. The main modules considered are amplifier and sampling, Discrete Wavelet Transform (DWT), quantization and encoding of the quantized DWT coefficients [13]. Since the processing time is tiny, we can ignore the processing time compared to the transmission time. We focus on optimizing the performance of transmission phase II. While the communication between the PDA and the central server is performed through WiFi following the IEEE 802.11 (WiFi) standard [14].

B. Encoder Energy Calculation

According to our work in [15], the encoding energy consumption is evaluated as

$$E_s = E_{DWT} + E_Q \tag{1}$$

where E_{DWT} is the energy consumed in DWT and E_Q is the quantization-encoding energy consumption. Using threshold-based DWT, we can control in the number of output samples generated from DWT and thus the compression ratio of the DWT. The compression ratio is evaluated as $C_R = 1 - \frac{M}{N_s}$, where M is the number of output samples generated after DWT and N_s is the length of the original signal. The computational complexity that is defined as the number of computations needed in the compression process, for N_s dimensional EEG signal, is calculated as

$$C_{DWT} = F \cdot N_s \sum_{l=0}^{l=L} \frac{1}{2^l} \tag{2}$$

where L is the number of decomposition levels and F is the wavelet filter length of the utilized wavelet family that is obtained as $F = 2\kappa$, where κ is the wavelet family order. Using this computational complexity, the energy consumed in the DWT-based encoding can be evaluated as

$$E_{DWT} = C_{DWT}(F, N_s, L) \cdot E_{comp} \tag{3}$$

where E_{comp} is the energy consumed per computation. For the sampling, quantization and encoding, the energy consumption depends on the number of conversion steps (to convert the input samples into bits), which in turn depends on the number of input samples to the quantization and encoding modules. Hence, the number of bits generated from sensor node i, after DWT and quantization, is calculated as

$$l_{comp} = N_s \cdot (1 - C_{Ri}) \cdot n = l_i \left(1 - C_{Ri}\right) \tag{4}$$

where n is the number of bits/sample. From [15], the encoding energy consumption is evaluated as

$$E_s = F \cdot N_s \left(\sum_{l=0}^{l=L} \frac{1}{2^l} \right) \cdot E_{comp} + N_s \left(1 - C_R\right) \cdot E_{CS} \tag{5}$$

where E_{CS} is the energy consumption at each conversion step in sampling and quantization process.

C. Encoder Distortion Calculation

The encoding distortion is measured by the Percentage Root-mean-square Difference (PRD) between the recovered EEG data and the original one. Using our real-time implementation that will be discussed later, we got the following relation between the encoding distortion and the compression ratio, as shown in Figure 2. Using this relation and by applying regression analysis, the encoding distortion D_s can be approximated as

$$D_s = c_1 e^{(1-C_R)} + c_2 \cdot (1 - C_R)^{-c_3} + c_4 \cdot F^{-c_5} - c_6 \tag{6}$$

where the model parameters c_1, c_2, c_3, c_4, c_5 and c_6 are estimated by the statistics of the typical EEG encoder used. A comparison between the encoding distortion of the real-time implementation and the encoding distortion calculated using regression analysis is presented in Figure 2.

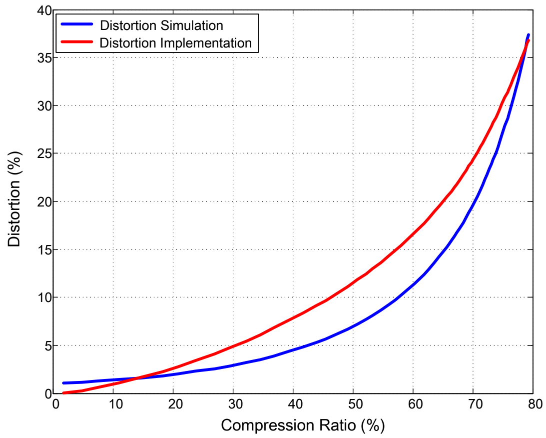

Fig. 2. The relation between the encoding distortion and the compression ratio.

D. Transmission Time Calculation

In this section, we will calculate the transmission time needed for the PDA to send N_p packets of compressed data to the server using IEEE 802.11 (WiFi) standard [14]. In the current Carrier Sense Multiple Access (CSMA) implementation in WiFi devices for a packet transmission, the PDA should first carrier senses the wireless channel; if the channel is idle, it sends the packet directly; otherwise, it randomly selects a waiting time period within $[0, CW]$ and senses the channel again. Accordingly, the average back-off time for one packet transmission can be approximated as

$$t_{cs} = \times \frac{CW}{2} \times \min \{(M-1)/2, R\} \tag{7}$$

where $CW = 640\mu s$ is the maximum time window size, $M-1$ is the number of potential contenders sharing the same bandwidth with the PDA, and R is the maximum number of back-off retries [16]. After the sensing stage, the time for the PDA to send a packet of data is calculated as

$$t_{tr} = \frac{S_h + S_d}{\theta} \tag{8}$$

where S_h and S_d are the sizes of header and payload of the WiFi packet, and θ is the maximum throughput of WiFi network [16]. Furthermore, we use p_r to specify the packet delivery ratios (PDR) of the data transmission from the PDA to the server, hence the expected number of transmissions for one successful packet delivery is $1/p_r$. Accordingly, the expected time needed for a successful delivery of one packet from the PDA to the server is $(t_{cs} + t_{tr})\, 1/p_r$. Accordingly, the transmission time needed for the PDA to send N_p packets of compressed data to the server is evaluated as

$$T_d = \sum_{i=1}^{i=N_p} \frac{t_{cs}(i) + t_{tr}(i)}{p_r(i)} \tag{9}$$

where

$$N_p = \frac{l_i\,(1 - C_{Ri})}{S_d} \tag{10}$$

is the number of data packets to be transmitted by the PDA after compression.

III. Energy Consumption Optimization

The objective of our optimization problem is to minimize the energy consumed by the PDA to compress and transfer its data to the server side under given delay and distortion constraints.

The proposed optimization problem can be formulated as a Energy-Delay-Distortion optimization problem, where the design objective is to minimize the processing energy under a delay deadline constraint and the maximum signal distortion at the receiver is below a certain distortion threshold. Therefore, the problem can be written as

$$\begin{aligned} &\min_{C_{Ri}, F_i} \; (E_s) \\ &\text{such that} \\ &\qquad T_d \le \tau_m \\ &\qquad D_s(i) \le D_{th}(i) \\ &F_i \le F_{max}, 0 \le C_{Ri} \le 1,\ \forall i \in N. \end{aligned} \tag{11}$$

This optimization problem is a function of the compression ratio C_{Ri}, wavelet filter length F_i, data length l_i, application layer distortion threshold D_{th} and maximum delay deadline τ_m. In our model, we aim to find the optimal encoder's parameters that decrease the complexity of the compression process and minimize the processing energy consumption. These parameters will change dynamically according to the channel state to satisfy delay and distortion constraints. Consequently, the unknowns in this optimization problem are the compression ratios C_{Ri} and wavelet filter lengths F_i.

By checking the operations that preserve convexity [9], this initial form of the optimization problem is not convex. Therefore, we opt to manipulating the original optimization problem to convert it into a convex Geometric Program as follows. First, let $1 - C_R = \overline{C_R}$.

Then, define $\widehat{C_{Ri}} = \log(\overline{C_{Ri}})$ and $\widehat{F_i} = \log(F_i)$. After that, substitute in (5) and (6). The function $\log(\sum_i \delta_i e^{a_i \cdot y_i})$ is convex if $\delta_i > 0, y_i \in \mathbb{R}$ [9]. Composition with an affine mapping preserves convexity. Using this approach, we will have

$$E_s = e^{\widehat{F_i}} \cdot N_s \left(\sum_{l=0}^{l=L} \frac{1}{2^l} \right) \cdot E_{comp} + N_s e^{\widehat{C_{Ri}}} \cdot E_{CS}. \tag{12}$$

Accordingly, the function E_s is convex in C_{Ri} and F_i.

The variables of this optimization problem will be $\widehat{C_{Ri}}, \widehat{F_i}$, for $i \in \{1, \cdots, N\}$. The number of variables grows as $2N$ and the number of constraints grows as $4N$. There are efficient interior-point methods to solve such problems [9].

IV. Real-Time Implementation

A. Implementation Framework

In this section, we propose the development and verification of a real-life application scenario to demonstrate the benefits of using our adaptive compression technique. Figure 3 presents the overall framework used. In this framework, the EEG data is collected from a patient using EEG Headset, then it is sent to a PDA (which will be a Smart Phone) that compresses the gathered data and forwards it to the server. At the server side, the reconstruction and distortion evaluation are performed to estimate the reconstruction accuracy. Furthermore, the EEG feature extraction, selection and classification are performed to detect the status of the patient.

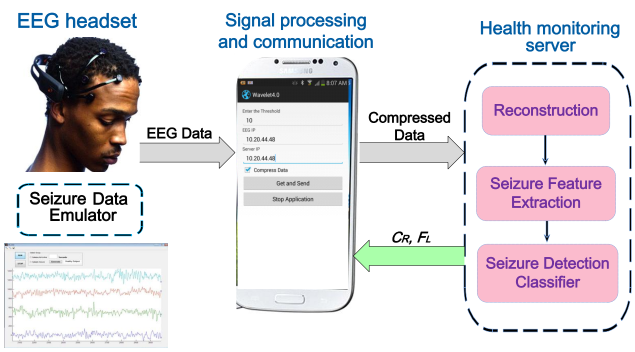

Fig. 3. Real-time Implementation Framework.

The main components of the proposed framework are:

- EEG Headset: Emotiv device is used as the EEG headset to collect data from the patient and send it to the PDA [17]. As it is difficult to find a patient with a seizure, we replaced the headset with a data emulator (which we named the client) to generate three classes: Healthy subject, subject with seizure but not active, and subject with active seizure.
- PDA (i.e. Smart Phone): We use the PDA as a processing unit to perform an adaptive compression on the received data from the data emulator.
- Health monitoring server: A server application is running to perform a real-time EEG classification on the received data from the PDA.

In this implementation framework, we extend our previous work in [18]. However, the main contributions of this work are:

- The data emulator is extended to support the communication with multiple PDAs.
- The PDA support the communication with the server through two TCP/IP sockets, one to send the data to the server, and the other one to receive a feedback from the server with the updated compression parameters (i.e. Compression Ratio and Filter Length).
- Updating the server side application to support the communication with multiple PDAs, which provides the flexibility of testing different scenarios for network users.

Figure 4 shows the android program used at the PDA. This program has three main functionalities: communicate with the data emulator, apply the adaptive compression technique on the received data, and forward the compressed data to the server. In this framework, we have the capability to operate different PDAs with different configurations at the same time, which enabled us to test different scenarios at the same time.

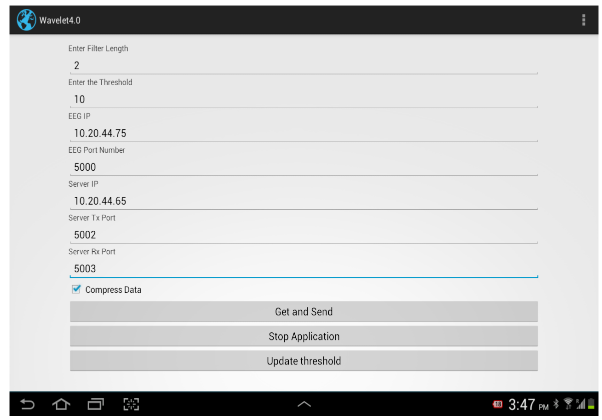

Fig. 4. Wavelet Android Application.

The compression process on the PDA is performed using DWT, where the compression ratio and filter length are updated using the feedback from the server side. In our setup, we are more concerned with the communication between the PDAs and the server. We used MATLAB software in both client and server sides. While the Java language is used in communication and compression processes on the Android platform (i.e. PDA). The communication between client, server and PDAs is performed through WiFi following the IEEE 802.11 (WiFi) standard [14].

B. Implementation of Enenrgy-aware Compression Algorithms

An Energy-aware adaptive compression algorithms are performed on the server side using optimization algorithms to update the encoder's parameters (C_{Ri} and F_i), for each PDA, subject to delay and distortion constraints. Three compression algorithms are considered in our implementation framework: Optimized Adaptive Compression (OAC), Iterative Adaptive Compression (IAC), and Maximum Compression (MC).

1) Optimized Adaptive Compression Algorithm (OAC):

OAC algorithms uses the optimization problem described in (11) to get the optimal encoder's parameters. Algorithm (1) describes the main steps of the OAC algorithm. First, the estimated delay time from the PDA to the server T_d is calculated by calculating the difference between the new data received from the PDA and the old one. In our implementation, we assumed that the data emulator continuously send the gathered data to the PDA. Using T_d, the value of p_r can be calculated using (9). After that, the optimization problem (11) is solved to get the optimal C_{Ri}'s and F_i's. Finally, the feedback is sent to the PDA with the calculated C_{Ri} and F_i. We should take into consideration that, due to practical constraints, the wavelet filter length must be an even number (e.g. 2, 4, 6 ... F_{max}) [19]. Therefore, we will round the calculated value of wavelet filter length to the nearest-higher value.

Algorithm 1 Optimized Adaptive Compression Algorithm (OAC)

1: Calculate T_d.
2: Find p_r using (9).
3: Solve the optimization problem in (11).
4: Calculate the optimal C_{Ri}'s and F_i's.
5: Send the new C_{Ri}'s and F_i's to the PDA.
6: End

2) Maximum Compression Algorithm(MC):

This algorithm does not perform any optimization process, it always sends the maximum allowable C_{Ri}'s and F_i's to the PDA, where the maximum value of F_i is F_{max} and the maximum C_{Ri} is calculated using (6). This value of C_{Ri} should satisfy the distortion constraint.

3) Iterative Adaptive Compression Algorithm (IAC):

In this algorithm, instead of solving the optimization problem in (11), we will use the iterative solution shown in Figure 5. The algorithm starts by setting the filter length to 2, which is the minimum value of F_i. Then, it calculates C_{Ri} using (6) such that the distortion constraint should be satisfied. After that, it checks the delay constraint. If the condition is not satisfied, it will increment the value of F_i and estimate again the corresponding C_{Ri}. This loop will continue till it gets the values of C_{Ri} and F_i that satisfy the delay and distortion constraints. In case of these two constraints cannot be satisfied together, due to bad channel conditions, the algorithm will set the values of C_{Ri}'s and F_i to the maximum allowable values, as in MC algorithm.

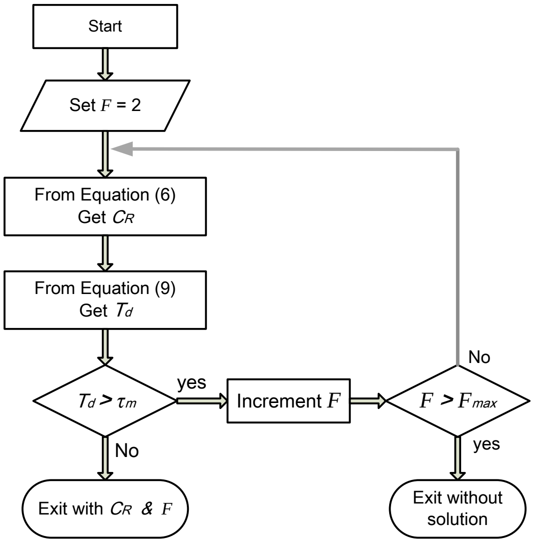

Fig. 5. IAC algorithm flowchart.

V. Performance Evaluation

A. Evaluation Setup

The simulation results were generated using the environment setup shown in Figure 6 and the technical requirements of selected BAN application [20]. Data acquisition is performed through a MATLAB code designed to communicate with an Emotive device [17] through the Bluetooth connection of the device. The MATLAB code can be defined to capture data for a certain amount of time from the user. The amount of data to be captured depends on the required features for correct classification process. It is assumed that there are three PDAs, each one communicates with the data emulator to collect the data. The data emulator collects a batch of data that corresponds to 25 channels (sensors), where each channel can capture 128 sample per second. We collect 32 seconds that corresponds to 4096 samples of epileptic EEG data [21]. This data (i.e. 25 x 4096 samples) is sent to each PDA to perform DWT compression. Then, the PDA sends the compressed data to the server, where the server performs optimization process according to the compression algorithms discussed in section IV-B to update compression parameters (i.e. Compression Ratio and Filter Length) according to the channel variations for each PDA. Then, the new compression parameters are sent to each PDA. Furthermore, at server side, the EEG feature extraction, classification and distortion evaluation are performed to detect the status of the patients. The support vector machine (SVM) classifier is used for the purpose of feature-based classification [22]. The simulation parameters used are given in Table I.

Fig. 6. Implementation environment setup.

B. Implementation Results

In this section, results from our real-time framework are presented. A comparison between the aforementioned compression algorithms is demonstrated through our real-time

TABLE I
IMPLEMENTATION PARAMETERS

Parameter	Value	Parameter	Value
E_{comp}	8 nJ	E_{CS}	21 nJ/step
F_{max}	20	BER	10^{-4}
L	2	N_s	4096 sample
D_{th}	10 %	n	12 bps
c_1	1.48	c_2	4.35
c_3	1.46	c_4	2.4
c_5	0.18	c_6	9.5
s_d	140 bytes	s_h	46 bytes
τ_m	1 sec	θ	54 $Mbps$
CW	640 μs	R	5

implementation to demonstrate the advantages of using the proposed adaptive compression algorithm. In Figure 7, a comparison between the implemented compression algorithms is demonstrated based on the saving time (δ) in the end-to-end delay that is defined as $\delta = T_t^{Nocomp} - T_t^{comp}$, where T_t^{Nocomp} is the total end-to-end delay in the system without applying any compression algorithm at the PDAs and T_t^{comp} is the total end-to-end delay in the system with compression. The time index is the number of received data batches with the time.

It can be shown that, MC algorithm defines the upper limit of (δ), as it uses the maximum compression's parameters. On the contrary, it works with maximum complexity and maximum energy consumption, as in Figure 8. Moreover, this algorithm always uses fixed compression's parameters whatever the channel state was. On the other hand, IAC and OAC algorithms are competing in their performance, where compression's parameters are adaptively changed according to the channel state at each PDA. These algorithms satisfy the delay and distortion constraints and achieve better performance than the no compression case. Furthermore, they minimize the complexity and energy consumption, as shown in Figure 8.

Figure 7 also illustrate that the performance of the IAC and OAC algorithms is almost the same. The reason of that is the filter length approximation, because the wavelet filter length must be an even number (e.g. 2, 4, 6 ... F_{max}) [19]. Therefore, in the OAC algorithm, we round the optimal-calculated value of the wavelet filter length to the nearest-higher value. This leads to decreasing the performance deviation between IAC and OAC algorithms. The only change in the saving time (δ) comes from the different channel conditions at each PDA, which leads to different number of retransmissions.

Figures 9 and 10 present the adaptive change of the C_{Ri} and F_i with channel conditions for the implemented compression algorithms using real-time transmission from three PDA. As shown in the figures, the MC algorithm always uses the maximum allowable C_{Ri}'s and F_i's. While both IAC and OAC algorithms find the best C_{Ri}'s and F_i that minimize encoder's complexity and energy consumption, and adaptively change them according to channel variations.

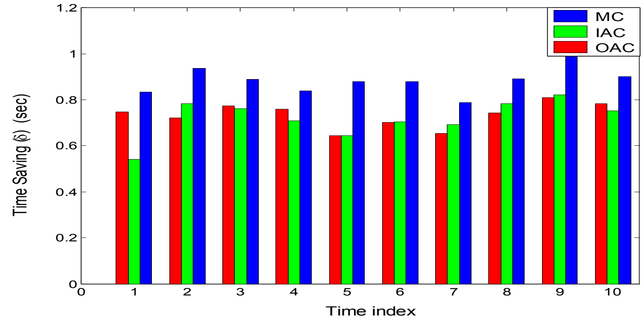

Fig. 7. Saving time of the implemented compression algorithms while varying the channel.

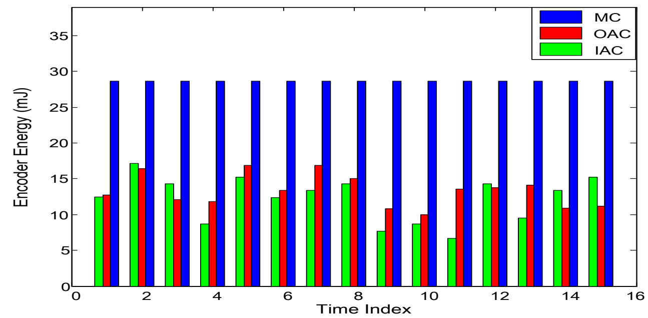

Fig. 8. Encoder Energy of the implemented compression algorithms while varying the channel.

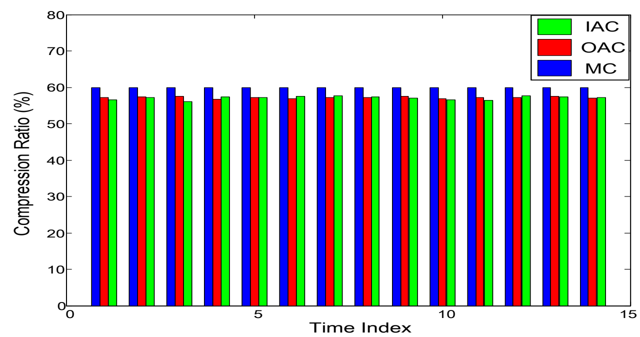

Fig. 9. Compression Ratio of the implemented compression algorithms while varying the channel.

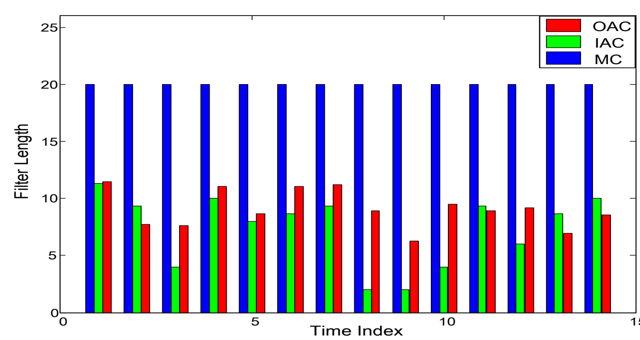

Fig. 10. Filter Length of the implemented compression algorithms in the varying channel.

VI. Conclusion

In this paper, an Energy-aware adaptive compression technique that aims at achieving low-complexity and energy-efficient compression is proposed. In designing this compression technique, we take into consider the time delay and distortion constraints of the considered application. In this context, we analyze the processing energy consumption, then an energy consumption optimization model with constraints of distortion and delay is proposed. Using this model, at the server side, we dynamically chooses the optimal compression parameters according to real-time measurements of the packet delivery ratio (PDR) or individual users that is highly depends on the channel state. Then, the server send the new compression parameters to the PDA to update its encoder before the next transmission. To evaluate and verify our compression technique, we develop an experimental testbed, where the EEG data is sent to the PDA that compresses the gathered data and forwards it to the server which decompresses and reconstructs the original signal and send feedback with the new encoder's parameters to the PDA. Experimental testbed and simulation results show that our adaptive compression technique can offer about 20% savings in the End-to-End delay with low complexity and without affecting application accuracies.

References

[1] A. Panayides, M. Pattichis, and C. Pattichis, "Mobile-health systems use diagnostically driven medical video technologies [life sciences]," *Signal Processing Magazine, IEEE*, vol. 30, no. 6, pp. 163–172, Nov 2013.

[2] O. Omeni, A. C. W. Wong, A. J. Burdett, and C. Toumazou, "Energy efficient medium access protocol for wireless medical body area sensor networks," *IEEE Transactions on Biomedical Circuits and Systems, VOL. 2, NO. 4*, Dec. 2008.

[3] M. Alam, O. Berder, D. Menard, and O. Sentieys, "Latency-energy optimized mac protocol for body sensor networks," pp. 67–72, May 2012.

[4] B. Otal, L. Alonso, and C. Verikoukis, "Highly reliable energy-saving MAC for wireless body sensor networks in healthcare systems," *IEEE Journal on Selected Areas in Communications, VOL. 27, NO. 4*, May 2009.

[5] Q. Shen and W. Zhuang, "Energy efficient scheduling for delay constrained communication in wireless body area networks," pp. 262–267, Dec 2012.

[6] F. Chiti, R. Fantacci, and S. Lappoli, "Contention delay minimization in wireless body sensor networks: A game theoretic perspective," in *Global Telecommunications Conference (GLOBECOM)*, Dec 2010, pp. 1–6.

[7] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw., VOL. 7, NO. 3, pp. 537568*, May 2009.

[8] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali, "Distributed cross-layer optimization in wireless networks: A second-order approach," *Proceedings in IEEE INFOCOM*, 2013.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. cambridge university press, 2003.

[10] M. Grant and S. Boyd, "cvx users guide for cvx version 1.21," *2011*, 2011.

[11] H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr, "A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy," *IEEE Transactions on Biomedical Engineering, VOL. 54, NO. 2*, Feb. 2007.

[12] R. Kottaimalai, M. P. Rajasekaran, V. Selvam, and B. Kannapiran, "EEG signal classification using principal component analysis with neural network in brain computer interface applications," *IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, Mar. 2013.

[13] R. Hussein, A. Awad, A. El-Sherif, A. Mohamed, and M. Alghoniemy, "Adaptive energy-aware encoding for DWT-based wireless EEG telemonitoring system," *IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pp. 245–250, Aug 2013.

[14] IEEE, "wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Standard 802.11*, 2007.

[15] A. Awad, R. Hussein, A. Mohamed, and A. A. El-Sherif, "Energy-aware cross-layer optimization for EEG-based wireless monitoring applications," *IEEE Conference on Local Computer Networks (LCN)*, 2013.

[16] Y. Li, X. Qi, M. Keally, Z. Ren, G. Zhou, D. Xiao, and S. Deng, "Communication energy modeling and optimization through joint packet size analysis of BSN and WiFi networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1741–1751, Sept 2013.

[17] "http://emotiv.com/."

[18] A. Awad, M. Hamdy, and A. Mohamed, "Transmission delay minimization for energy constrained communication in wireless body area sensor networks," in *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, March 2014, pp. 1–5.

[19] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed. Academic Press, 2008.

[20] B. Z. et al., "TG6 technical requirements document," *IEEE P802.15-08-0644-09-0006; https://mentor.ieee.org/802.15/.*

[21] R. Andrzejak, K. Lehnertz, C. Rieke, F. Mormann, P. David, and C. Elger, "Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, 64, 061907*, 2001.

[22] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classication," *Department of Computer Science, National Taiwan University, Tech. Rep., [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/papers.html*, 2003.